Group Allocation Multiple Access with Collision Detection

Andrew Muir **J.J. Garcia-Luna-Aceves**
Computer Engineering Department
University of California
Santa Cruz, CA 95064

Abstract

The Group Allocation Multiple Access with Collision Detection (GAMA/CD) protocol for scheduling variable-length packet transmissions in a local area network is specified and analyzed. GAMA/CD provides the advantages of both TDMA and CSMA/CD by maintaining a dynamically-sized cycle that varies in length depending on the network load; each cycle is composed of a contention period and a group-transmission period. During the contention period, a station with one or more packets to send competes for membership in the transmission group. Once a member of the transmission group, a station is able to send data without collision during each cycle; as long as a station has data to send, it maintains its position in the group. This can be viewed as either allowing stations to "share the floor" in an organized manner, or as establishing frames that are not synchronized on a slot-basis and vary their length dynamically based on demand. Both the throughput and the delay of GAMA/CD are presented and analyzed. To validate our analysis, the results of both models are compared to the throughput and delay produced by a simulation of GAMA/CD.

1 Introduction

Such medium access control (MAC) protocols as CSMA/CD [10] that require a station to contend for the ability to send each data packet of a message cannot provide performance guarantees. This is a significant problem for real-time multimedia applications requiring long-term connections and bounded jitter. Many strategies have been proposed that can provide some form of performance guarantees in the MAC protocols of local area networks; these include: fixed assignment (e.g., TDMA, FDMA), polling, token passing, and dynamic reservation protocols. However, each of these strategies can be improved in terms of its performance or implementation complexity.

This work was supported in part by DARPA under Grant DAAB07-95-C-D157.

In TDMA, the average delay experienced by a station that has been assigned a data slot is constant, regardless of the channel load; this is good if the load is high, but if the load is low the delay is longer than necessary, unless data slots can be quickly reallocated to the active stations. Polling requires a central station to direct the transmission of the other stations, and wastes polling time when the majority of stations are idle. A token passing network does not require a central station, but must deal with cases in which the token is lost or duplicated, which increases the complexity of the protocol. Dynamic reservation protocols are based on the premise that "control frames" can be defined in which stations can reserve the right to use slots in "data frames," the length of the control frames may be fixed or variable, but it is a function of the number of stations in the system. A control frame whose size is defined by the number of stations complicates the addition and deletion of stations and does not scale very well. Much work has been done in the area of real-time data transmission across a multi-access LAN. In [7] and [13] window protocols are decribed which require that stations are synchronized in order to support time slotting. In [4] a protocol similar to GAMA/CD is described; however, this protocol requires time synchronization, and the frame size does not vary with the level of network traffic. In [1] a protocol that transmits voice packets over virtual circuits is described; voice traffic is given a higher priority than data traffic. Other schemes which use a token passing scheme are described in [11], [12] and [14].

We describe and analyze a new MAC protocol for LANs, which we call Group Allocation Multiple Access with Collision Detection (GAMA/CD). GAMA/CD provides dynamic reservations of the channel and its implementation complexity is comparable with that of CSMA/CD protocols.

GAMA/CD builds a dynamically-sized "cycle" that grows and shrinks depending upon traffic demand.

Report Documentation Page						*Form Approved* *OMB No. 0704-0188*
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.						
1. REPORT DATE **1997**		2. REPORT TYPE				3. DATES COVERED **00-00-1997 to 00-00-1997**
4. TITLE AND SUBTITLE **Group Allocation Multiple Access with Collision Detection**						5a. CONTRACT NUMBER
						5b. GRANT NUMBER
						5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)						5d. PROJECT NUMBER
						5e. TASK NUMBER
						5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064**						8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)						10. SPONSOR/MONITOR'S ACRONYM(S)
						11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**						
13. SUPPLEMENTARY NOTES						
14. ABSTRACT						
15. SUBJECT TERMS						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON	
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**9**		

Each cycle consists of a contention period of up to a maximum duration that is independent of the number of user stations, and a group-transmission period during which one or more stations transmit data packets without collisions. A position in the transmission group is allocated to an individual station, and a station can continue to transmit in this position as long as it has data to send.

We believe that GAMA/CD combines the best features of CSMA/CD and contention-free protocols like TDMA. On the one hand, like CSMA/CD, GAMA/CD is very efficient under light load. On the other hand, GAMA/CD is much more stable under heavy loads than CSMA/CD, because it permits stations in the transmission group to send packets independently of new requests for additions to the transmission group. GAMA/CD ensures that, once a station has reserved a position in the group-transmission period, it will be able to transmit at or better than a guaranteed rate. The guaranteed rate is given by the maximum length of any given cycle, which is the sum of a maximum contention period and a maximum transmission period. A maximum contention period is directly proportional to the maximum propagation delay over the LAN, is independent of the number of stations, and is very short in high-speed LANs. Therefore, just like in other schemes (e.g., token ring schemes) the time between two transmissions by the same station is bounded.

Section 2 describes GAMA/CD in detail; section 3 analyzes the throughput of GAMA/CD. Section 4 studies its average delay using an approximate model. Section 5 compares the results of our analytical models to simulated results. Finally, section 6 offers some concluding remarks.

2 Protocol Description

GAMA/CD divides the channel into a series of cycles; each cycle consists of a contention period and a group-transmission period. The group-transmission period is further divided into a set of individual transmission periods; the number of individual transmission periods per cycle varies with the amount of network traffic. An individual transmission period corresponds to a slot in a synchronized network; however, GAMA/CD does not require that network stations be synchronized by time. During the contention period stations contend for membership in the "transmission group"; each group member is able to transmit data without collision during the tranmission period allocated to the station when it joined the transmission group. GAMA/CD uses a form of "limited sensing" as in [3] to allow a newly activated station to contend for group membership without knowing the entire state of the transmission group.

When a station receives a message to transmit, it listens to the channel for 2τ seconds, where τ is the maximum end-to-end propagation delay. As will be explained later in this section, the value 2τ is chosen in order to ensure that no data transmitted by a group member is involved in a collision. If the listening station does not detect a signal on the channel within the 2τ-second interval, it transmits the *initial packet* of the message. As the station transmits the *initial packet*, it also senses the channel for a possible collision. If a collision is detected (Fig. 2), the station stops transmitting; since the propagation delay is at most τ, any collision that occurs is detected within 2τ seconds of the start of transmission. After sensing the collision, the transmitting station sends a jamming packet; the purpose of this packet is to ensure that every station is aware of the collision. If the transmission of the *initial packet* is succesfully completed (Fig. 1), and the transmission group is not full, a transmission period is created and allocated to the new member.

The header of each transmission period contains the number of group members; thus, a new member is able to determine this value by reading the header of a transmission period. While a station is a group member, it is required to listen to each contention period, to determine whether or not a station has been added to the group. Each member is also required to listen to every transmission period; an idle transmission period means that a group member has either failed, or voluntarily left the group; in either case, the transmission period is removed. Because a group member listens to every contention and transmission period, it knows the number of group members, and its own position within the group.

The contention period begins with up to 3τ seconds of idle time; therefore, any message which arrives in the first τ seconds of the contention period will generate an *initial packet*. If the first station in the transmission group does not detect a signal 5τ seconds after the start of the contention period, the contention period must be idle, and the first station begins its transmission period; otherwise, the first station waits until the channel is clear before beginning its transmission. The first station must wait for up to 5τ seconds because it is possible that another station τ seconds away from the first station begins the contention period τ seconds after the first station. If a message arrives at this other station τ seconds into the contention period, an *initial packet* is transmitted 2τ seconds later, and arrive at the first station after another τ seconds.

An *initial packet* sent during the last τ seconds of a contention period collides with the first transmission period. Accordingly, the first station sends a 2τ-second jamming packet before it transmits its data packet; this packet will collide with any *initial packet* in the channel, and force the transmitting station to stop sending its *initial packet*. It is possible that an *initial packet* might be sent τ seconds after the first transmission period has begun; this *initial packet* will arrive up to τ seconds later. Therefore, the jamming packet must be at least 2τ seconds long to ensure that no data is involved in a collision. The jamming packet ensures that the length of the contention period is no greater than $3\tau + \delta$ (when an *initial packet* is successfully transmitted) where δ is the maximum length of the *initial packet*.

A group member begins transmitting data as soon as it senses the channel is clear following the reception of the previous transmission period. As the maximum propagation delay is τ seconds, the delay between successive transmission periods will be no longer than 2τ seconds. If a group member does not receive a signal within 2τ seconds, the transmission period must be idle (Fig. 3); when this happens, each group member transmits a τ second long jamming packet. The jamming packet ensures that any *initial packet* transmitted during the idle transmission period is not successful. If a station does not detect a signal on the channel within 3τ seconds after succesfully transmitting an *initial packet*, the transmission group must be empty; therefore, the station becomes the first member.

In order for member stations to provide quality of service guarantees, GAMA/CD ensures that the length of the group-transmission period is not larger than some value G; therefore, the interval between successive occurrences of a transmission period is bounded. When a station receives a message to transmit, it determines the portion of each cycle it requires in order to achieve its quality of service guarantees. A station assumes that the cycle length is maximized (equal to $3\tau + \delta + G$) when it calculates this value. When the the length of the cycle is small, the interval between the transmission period is also small; consequently, as the length of the cycle increases, so does the interval between successive occurrences of the transmission period. As this interval increases, a station is required to transmit more data during each cycle, in order to meet its quality of service guarantees. Therefore, the maximum cycle length is used to ensure that member stations have the required bandwidth.

The allocated length of each transmission period is contained within the transmission period's header. Each group member knows the length of the entire group-transmission period by reading each header; if the *initial packet* is transmitted without collision, the transmitting station must first read the cycle length from the header of a transmission period. If there is enough available bandwidth, the station is admitted; otherwise, it backs-off. Once the station is added to the group, it is able to send the remaining packets without contention in subsequent cycles; after a station has been added, it maintains its position until it has no data to transmit. A succesful RTS specifies how many packets a station that is added to the transmission group will be sending.

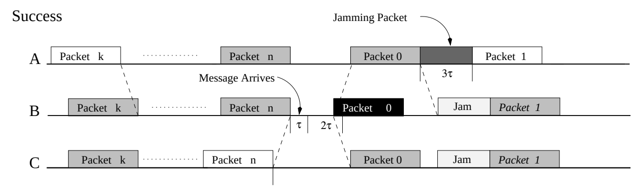

Figure 1: GAMA/CD: A successfully transmitted *initial packet*.

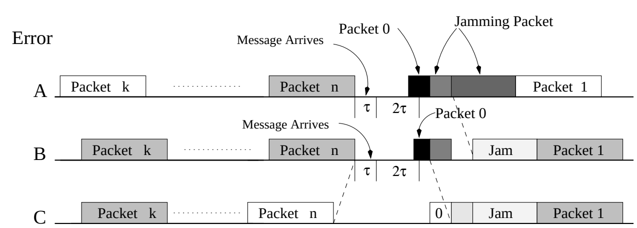

Figure 2: GAMA/CD: Two *initial packets* involved in a collision.

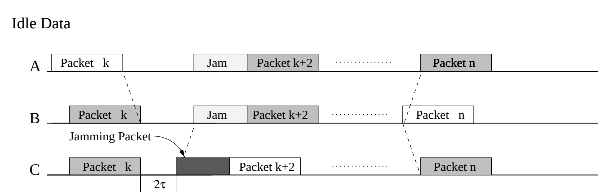

Figure 3: GAMA/CD: An idle transmission period.

3 Approximate Throughput Analysis

In the following analysis we develop a non-persistent model for the throughput of GAMA/CD. The model assumes that there is an infinite number of stations; the stations form a Poisson source generating RTSs (both new and retransmitted) at a mean arrival rate of λ, and each station is assumed to have at most one RTS to transmit at any time. The time

to transmit a single data packet is δ, the number of packets in a message is a random variable, and the probability that a message will complete its transmission (in a given cycle) is given by $\mu = \frac{1}{N}$ where N is the average number of packets in a message. The transmission channel does not introduce errors, any errors observed in received packets are the result of collisions, and collisions are detected by all stations. We assume that there are no station failures, that the propagation delay is τ between all stations, and that the members in the transmission group are ordered by the number of packets (in the message) remaining to be transmitted.

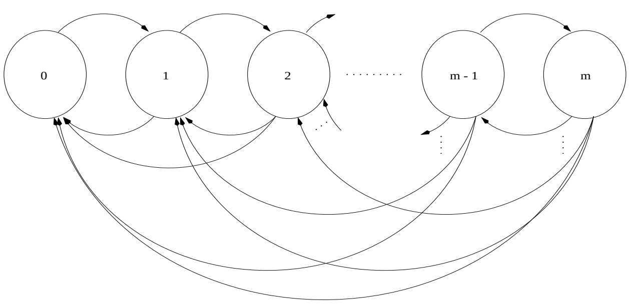

Figure 4: Markov Chain defining the average number of members in the group.

The state of the Markov chain in Fig. 4 represents the number of members in the group, where the value m is equal to the maximum number of group members. The probability that a station is added to the group $P_a(n)$ where n is the current number of group members, is equal to the probability that an *initial packet* is successfully transmitted within a contention period, which is the same as the probability that only one message arrives during the first τ seconds of the contention period. If there are m group members, no new members can be added because the transmission group is full. The value of $P_a(n)$ can be expressed as

$$P_{a(n)} = \begin{cases} \lambda\tau e^{-\lambda\tau} & \text{if } n < m \\ 0 & \text{if } n = m \end{cases} \tag{1}$$

The probability P_i that a contention period is idle is equal to the probability that no packets arrive during the first τ seconds of the contention period; Therefore,

$$P_i = e^{-\lambda\tau} \tag{2}$$

The probability P_e that a contention period results in an error is the same as the probability that more than one message is transmitted during the contention period. This value can be expressed as:

$$P_e = 1 - \lambda\tau e^{-\lambda\tau} - e^{-\lambda\tau} \tag{3}$$

If a line is drawn between any two states of the Markov chain, then the flow in one direction across the line has to equal the flow in the other direction. For an arbitrary state n where $0 \leq n < m$ the probability that the number of group members changes from n to $n+1$ is equal to the probability that the state of the Markov chain transitions from a state N (where $N > n$) to a state L (where $L \leq$ n). This can be represented by the following equation.

$$\begin{aligned} & P_n P_{a(n)}(1-\mu)^n \\ & = \sum_{i=1}^{m-n}(1-P_{a(n+i)})P_{n+i}\sum_{j=i}^{n+i}\mu^j(1-\mu)^{n+i-j} \\ & + \sum_{i=1}^{m-n} P_{a(n+i)}P_{n+i}\sum_{j=i+1}^{n+i}\mu^j(1-\mu)^{n+i-j} \end{aligned} \tag{4}$$

$$\begin{aligned} & P_n P_{a(n)}(1-\mu)^n \\ & = \sum_{i=1}^{m-n} P_{n+i}\sum_{j=i}^{n+i}\mu^j(1-\mu)^{n+i-j} \\ & - \sum_{i=1}^{m-n} P_{a(n+i)}P_{n+i}\mu^i(1-\mu)^n \end{aligned} \tag{5}$$

Dividing both sides of Eq. 5 by $(1-\mu)^n P_{a(n)}$ leads to

$$\begin{aligned} P_n & = \frac{1}{P_{a(n)}} \\ & * \sum_{i=1}^{m-n} P_{n+i}[\sum_{j=i}^{n+i}\mu^j(1-\mu)^{i-j} \\ & - P_{a(n+i)}\mu^i] \end{aligned} \tag{6}$$

where $0 \leq n < m$.

Successively substituting the values of P_{n+i} in Eq. 6 results in the following equation for P_n which is dependent only upon the value of P_m:

$$P_n = P_m F(n) \tag{7}$$

Where $F(n)$ is a recursive function which can be defined as follows:

$$\begin{aligned} F(n) = \sum_{i=1}^{m-n} F(n+i) \quad [& \frac{1}{P_{a(n)}}\sum_{j=i}^{n+i}\mu^j(1-\mu)^{i-j} \\ & - P_{a(n+i)}\mu^i] \end{aligned} \tag{8}$$

The sum $\sum_{n=0}^{m} P_n = 1$ Therefore,

$$P_m = \frac{1}{\sum_{i=0}^{m-1} F(i) + 1} \tag{9}$$

The average number of group members ρ is

$$\rho = \sum_{j=1}^{m} j P_j \tag{10}$$

An idle period separates successive cycles; this is either the period of time in which the transmission group is empty, or the idle period at the start of a contention period. When the transmission group is empty, i.e., no stations have packets to send, the idle period lasts $(\frac{1}{\lambda} + 2\tau)$ seconds (on average). The first term $(\frac{1}{\lambda})$ represents the time between Poisson message arrivals; the second term $(2\tau)$is the time a station must listen for a clear channel. If the number of group members is greater than 0, then the length of the idle period depends upon the result of the contention period. If the contention period is idle, the length is equal to 3τ; the first τ seconds are the period of time in which arriving messages can be successfully transmitted and, the following 2τ seconds represent the time a station must wait to sense for a clear channel. If the contention period is not idle, the channel is idle for a period of $\frac{5\tau}{2}$ seconds (on average). For an *initial packet* to be transmitted it must arrive within the first τ seconds of the contention period; on average a packet will arrive after $\frac{\tau}{2}$ seconds, and wait for 2τ seconds before the *initial packet* is transmitted. Therefore, the average length $\overline{I}$ of the idle period can be expressed as:

$$\overline{I} = P_0 \left(\frac{1}{\lambda} + 2\tau \right) + (1 - P_0) \left[P_i 3\tau + (1 - P_i)\frac{5\tau}{2} \right] \tag{11}$$

The probability that an *initial packet* (once sent) is successful P_r is equal to the probability that no other messages arrive within τ seconds of the start of transmission, i.e.,

$$P_r = e^{-\lambda\tau} \tag{12}$$

The average length $\overline{B}$ of a busy period is the length of a busy contention period plus the length of the group-transmission period. If there are no group members a busy period does not start until a message arrives at a station; therefore, when the number of group members is 0, a contention period will result in either a successful *initial packet* or in the collision of multiple *initial packets.* The probability that an *initial packet* is successful is equal to the probability that no other messages arrive within τ seconds of the arrival of the first message; this is equal to P_r, and the length of a successful contention period is $(\delta + \tau)$. Consequently, the probability that a collision occurs is equal to $(1 - P_r)$. When a contention period produces an error, the length of the contention period is equal to $\overline{Y} + 3\tau$. Where $\overline{Y}$ represents the length of the overlap between colliding *initial packets.* This value as shown in [6] can be represented as:

$$\overline{Y} = \tau - \frac{1 - e^{-\lambda\tau}}{\lambda} \tag{13}$$

The 3τ seconds represents the time before colliding stations realize a collision has occured plus the length of the jamming packet and the propagation delay for the jamming packet to reach the other stations. If the transmission group is not empty, the length of the busy period depends upon the outcome of the contention period; the length of the contention period is: $\delta + \tau$ if an initial packet is successfully transmitted, $\overline{Y} + 3\tau$ if multiple initial packets collide, and 0 if the contention period is idle. Regardless of this outcome, the length of the group-transmission period is $\rho(\delta+\tau)$. Therefore, the value of $\overline{B}$ can be expresses by the following equation:

$$\begin{aligned} \overline{B} \;=\; & P_0[P_r\tau + (1 - P_r)(\overline{Y} + 3\tau)] \\ & + (1 - P_0)[P_a(\rho)(\rho + 1)(\delta + \tau) + P_i\rho(\delta + \tau) \\ & + P_e[\overline{Y} + 3\tau + \rho(\delta + \tau)] + 3\tau] \end{aligned} \tag{14}$$

The average time spent transmitting data in a cycle $(\overline{U})$ is equal to the average number of group members plus the possibility that an *initial packet* is successful multiplied by the data packet length:

$$\overline{U} = (\rho + P_a(\rho))\delta \tag{15}$$

The throughput S is equal to the average time spent transmitting data in a cycle $(\overline{U})$ divided by the duration of an average cycle, i.e.,

$$S = \frac{\overline{U}}{\overline{I} + \overline{B}} \tag{16}$$

4 Average Delay Analysis

In this section the average delay of GAMA/CD is analyzed; the delay of a message is defined as the elapsed time from the instant the message is ready to be transmitted to the time the entire message is received at the destination. The average delay is modeled by the process shown in Fig. 5.

In order to determine the average delay, one must consider the steps a station takes in transmitting a message. When a station receives a message, it is in

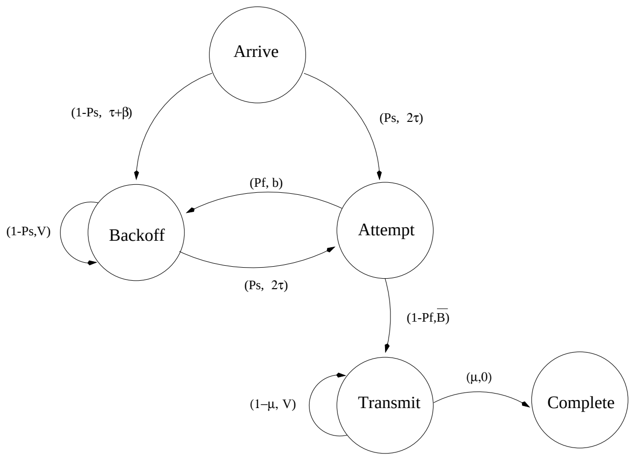

Figure 5: Process defining GAMA/CD delay characteristics.

the **Arrive** state. If the station senses the channel clear for 2τ seconds after the arrival of a message, it sends an *initial packet* (**Attempt**) otherwise it backs off. If the *initial packet* is successful the station enters the **Transmit** state, where it remains until the message is complete. If the *initial packet* is not successful the station enters the **Backoff** state.

These five states (Arrive, Backoff, Attempt, Transmit and Complete) along with the associated delay values and the transition probabilities define the process used to obtain an expression for the average delay. In Fig. 5, each link represents a transition from one state to another; for every link there is a label which consists of the probability the link is used, and the length of time spent in the state.

The transition from the **Arrive** state to the **Attempt** state is made with probability P_s, where P_s can be expressed as:

$$P_s = \frac{\overline{I}}{\overline{I} + \overline{B}}(1 - P_m) \tag{17}$$

P_s is the probability that a message arrives during an idle period and that the group is not full. Because a station must sense that the channel is clear for 2τ seconds before it sends the *initial packet*, the delay caused by the transition from **Arrive** to **Attempt** is 2τ. If a station is forced to back-off, it will have spent (on average) τ seconds listening to the channel before it detects a signal. The length of time a station backs-off is a random variable with an average value of b.

After waiting in the **Backoff** state, a station will enter the **Attempt** state with probability P_s; while in the **Attempt** state, a station sends an *initial packet* which has a probability of failure equal to P_f, where $P_f = 1 - e^{\tau\lambda}$. If the *initial packet* is successful, the station transitions to the **Transmit** state; the delay accrued by this transition is equal to $\overline{B}$. If the *initial packet* fails, the **Backoff** state is entered; in this case, the additional delay is equal to b. While in the **Transmit** state, a station sends a packet each cycle; the probability that a station completes its message transmission (in a given cycle) is equal to μ. Each time a station returns to the **Transmit** state, the additional delay is equal to the time required to transmit a complete cycle.

From Fig. 5 one can obtain an expression for the average delay D:

$$D = (1 - P_s)(\tau + b + E(B)) + P_s(2\tau + E(A)) \tag{18}$$

Where E(A) is the additional delay accumulated each time the **Attempt** state is entered, and E(B) is the delay caused by each stay in the **Backoff** state. The value for E(A) can be expressed as follows:

$$E(A) = (1 - P_f)(\overline{B} + E(T)) + P_f(b + E(B)) \tag{19}$$

where E(T) is equal to the time spent in the **Transmit** state. The expression for E(B) is

$$E(B) = P_s(2\tau + E(A)) + (1 - P_s)(b + E(B)) \tag{20}$$

solving for E(A) leads to:

$$E(A) = E(B) - \frac{1 - P_s}{P_s}b - 2\tau \tag{21}$$

and the equation for E(T) is as follows:

$$E(T) = (1 - \mu)(V + E(T)) \tag{22}$$

which simplifies to

$$E(T) = \frac{(1 - \mu)V}{\mu} \tag{23}$$

$$E(T) = \sigma V \tag{24}$$

Where $\sigma = \frac{(1-\mu)}{\mu}$, and V is the length of the entire cycle which can be expressed as follows:

$$V = \overline{I} + \overline{B} \tag{25}$$

Substituting Eq. 21 into Eq. 19 leads to:

$$E(B) - \frac{1 - P_s}{P_s}b - 2\tau = (1 - P_f)(\overline{B} + E(T)) + P_f(b + E(B)) \tag{26}$$

$$E(B) = \overline{B} + \sigma V + \frac{P_f}{1-P_f}b + \frac{1-P_s}{P_s(1-P_f)}b + 2\tau \quad (27)$$

Using Eq. 21 leads to the following expression for E(A):

$$E(A) = \overline{B} + \sigma V + \frac{P_f}{1-P_f}b + \frac{1-P_s}{P_s(1-P_f)}b + 2\tau - \frac{1-P_s}{Ps}b \quad (28)$$

Substituting the previous two equations into Eq. 18 leads to:

$$D = \overline{B} + \sigma V + \frac{P_f}{1-P_f}b + \frac{1-P_s}{P_s(1-P_f)}b + 2\tau + (1-Ps)b \quad (29)$$

5 Performance Results

The graphs presented in this section are based on networks with a speed of 100 Mb/s, and a distance between stations of 200 meters. For all graphs the unit of time is equal to the length of the end-to-end propagation delay.

Fig. 6 plots the throughput and delay of GAMA/CD; each curve represents a different value for the average number of packets in a message. The packet size is fixed at 200 bytes, but the average message length is varied by changing the number of packets per message. When the average number of packets per message is high (200 packets) the throughput rises to its maximum value immediately because each successful *initial packet* is able to reserve a collision-free transmission period for 200 packets (on average). Even when the average number of packets is low, GAMA/CD is able to produce a high throughput level over a broad range of network loads. As the number of packets per message increases so does the number of cycles each station spends in the group, which increases the average group size. Because the group size is larger, the delay between successive transmissions from a single station is longer; thus, the average message delay is increased.

In Fig. 7 we show the effect of changing the maximum size of the transmission group on the throughput and delay. When the number of group members is large, the overhead of the contention period is shared among a greater number of transmission periods; therefore, a larger fraction of time is spent transmitting data, and the throughput is increased. However, because the number of group members is greater; the size of the cycle increases, which leads to an extremely high delay.

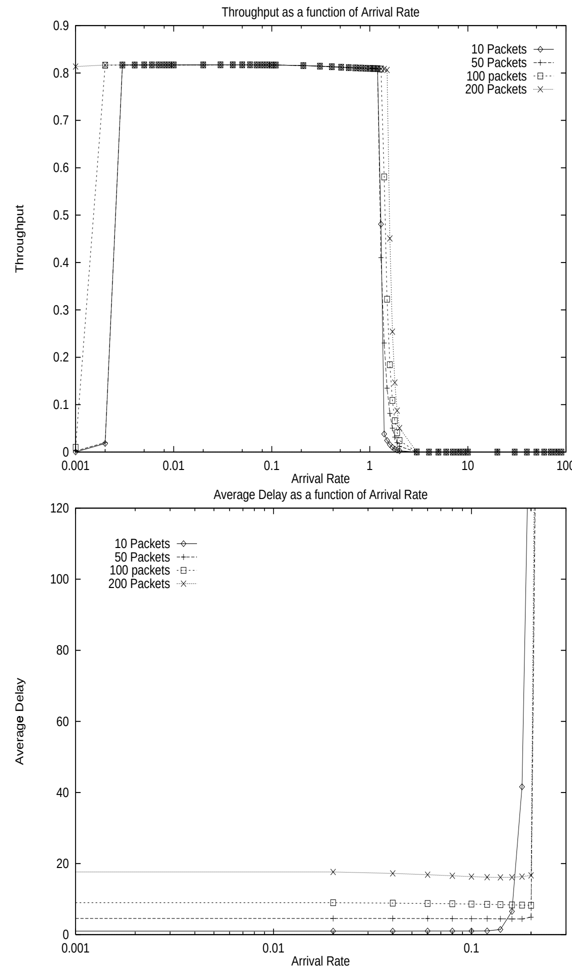

Figure 6: GAMA/CD: A comparison of the approximate throughput and average delay the arrival rate is allowed to vary. The average number of packets per message varies from 10 to 200, and the maximum number of group members is 50

Fig. 8 compares the throughput produced by our analytical model to the observed throughput of a simulation of GAMA/CD. This comparison is made in order to ensure that the assumptions we made in deriving the analytical models are valid. The average packet size is 200 bytes, the average number of packets per message is 50, the maximum group size is 50. There are either 50 or 100 stations in the simulated network, and the analysis uses an infinite number of stations. In a network where the number of stations is close to the maximum number of group members, the arrival rate decreases as more stations join the group. Because the arrival rate decreases, the throughput and delay stabilize, in much the same way as TDMA; the throughput maintains the maximum level for a longer

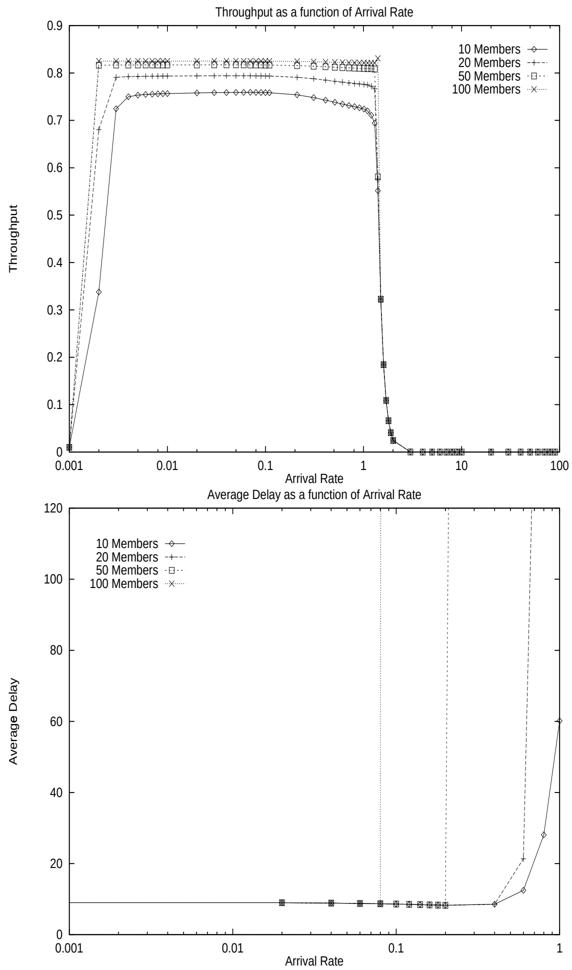

Figure 7: GAMA/CD: A comparison of the approximate throughput and average delay when the arrival rate varies. The average number of packets per message is 50, and the maximum number of group members varies between 10 and 100.

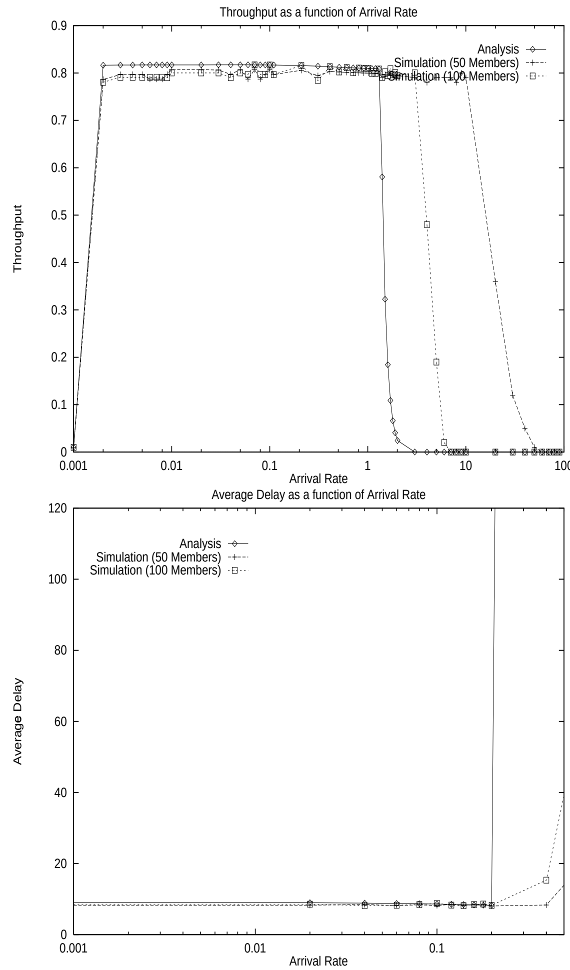

Figure 8: GAMA/CD: A comparison of the simulated and analytical throughput and average delay when the arrival rate varies. The average number of packets per message is 50, and the maximum group size is 50. The number of stations in the simulated networks is either 50 or 100.

period of time, and the increase in delay depends only upon the delay at the local node.

6 Conclusion

GAMA/CD is able to support quality of service guarantees by organizing the channel into dynamically sized cycles, each of which is composed of a contention period, and a group-transmission period. A station is allowed to contend for membership in the transmission group during the contention period; a member of the transmission group can transmit data collision free during each cycle. When the network load is light, GAMA/CD behaves much like CSMA/CD. As the network load rises, the number of group members increases to a pre-defined maximum, after which GAMA becomes in effect TDMA, giving every station that is part of the transmission group a "slot" in which to transmit. GAMA/CD can adapt to changing network conditions, even in the presence of sudden bursts of activity, and offers performance guarantees to stations that have been successfully added to the transmission group.

Our analytical results provide an approximation of the performance of GAMA/CD for the case in which stations have variable length messages to transmit; they also provide good insight into the effect various parameters (e.g. average packet size or network load) have on the performance of the protocol. Simulation results were used to validate the simplifying assumptions used in our derivation of the approximate

throughput and average delay. Our work continues to analyze the behavior of GAMA/CD after failures in the transmission group, and to introduce more sophisticated collision resolution mechanisms for *initial packets* within GAMA/CD to ensure that every cycle has a successful contention period.

References

[1] I. Chlamtac "An Ethernet Compatible Protocol for Real-Time Voice/Data Integration", *Computer Networks and ISDN Systems*, January 1985

[2] J. Hammond and P. O'Reilly "Performance Analysis of Local Computer Networks", Adison-Wesley 1986.

[3] P. Humblet "On the Throughput of Channel Access Algorithms with Limited Sensing", *IEEE Trans. on Communications*, April 1986.

[4] F. Jia and B. Mukherjee "The Superchannel Scheme for Integrated Services on Multiple Broadcast Networks", *Computer Networks and ISDN Systems*, October 1995.

[5] L. Kleinrock "Queuing Systems Volume 1: Theory", Wiley-Interscience 1975.

[6] L. Kleinrock and F. Tobagi "Packet Switching in Radio Channels: Part I-Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics", *IEEE Transactions on Communications*, December 1975.

[7] J. Kurose, M. Schwartz and T. Yemini "Controlling Window Protocols for Time-Constrained Communication in Multiple Access Environments", *Proc. IEEE Intl. Data Comm. Symposium*, 1993.

[8] R. Metcalf and D. Boggs "Ethernet: Distributed Packet Switching for Local Computer Networks", *Communications of the ACM*, July 1976.

[9] R. Rom and M. Sidi, Multiple Access Protocols Performance and Analysis, Springer-Verlag, 1990.

[10] F. Tobagi and V. Hunt "Performance Analysis of Carrier Sense Multiple Access with Collision Detection", *Computer Networks*, Vol. 4, October-November 1980.

[11] A. Valenzano, P. Montuschi and L. Ciminiera "Some Properties of Timed Token Medium Access Protocols", *IEEE Trans. on Software Engineering*, August 1987.

[12] C. Venkatramani and T. Chiueh "Design, Implementation, and Evaluation of a Software-Based Real-Time Ethernet Protocol", *Proc. ACM SIGCOMM '95*, 1995.

[13] W. Zhao, J. Stankovoc and K. Ramamritham "Virtual Time CSMA Protocols for Hard Real-Time Communications", *IEEE Trans. on Software Engineering*, August 1987.

[14] W. Zhao, J. Stankovoc and K. Ramamritham "A Window Protocol for Transmission of Time Constrained Messages", *IEEE Trans. on Computers*, September 1990.